# CS 2336 – Lab 2 Assignment

## Pre---Lab (10 Points)

Write pseudocode for problems 1 and 2 below. Describe step by step how you will construct the tallies for problem 1 as well as how you will sort the list of numbers and find the middle one for problem 2. Hand write (or type and print) your pseudocode and submit with your assignment.

---

## Magic 8 Ball (25 points)


Write a Java program that prompts the user to ask a question, and a random reply will be displayed.

Use the Random class to generate a random number to randomly choose one of the following 20 responses from a String array:

- It is certain
- It is decidedly so
- Without a doubt
- Yes definitely
- You may rely on it
- As I see it, yes
- Most likely
- Outlook good
- Yes
- Signs point to yes
- Reply hazy
- Ask again later
- Better not tell you now
- Cannot predict now
- Concentrate and ask again
- Don't count on it
- My reply is no
- My sources say no
- Outlook not so good
- Very doubtful

After a question has been answered, prompt for another question. Repeat continuously until the user presses the Return key without a question. When this happens, exit the loop and exit the program.

***Sample Output:***

```
> java Magic8Ball
I AM THE MAGIC 8 BALL

Ask me a question about your future: Will I be rich?

Concentrate and ask again

Ask me a question about your future: Will I be wealthy?

Reply hazy

Ask me a question about your future: Will I marry the person of my dreams?

Signs point to yes

Ask me a question about your future:
>
```

## Tip Calculator (25 points)


Create a class called TipCalculator which calculates a tip amount based on a grade of service received.

Create a method named ***calculateTip*** which accepts two incoming parameters: a double and a String. The first parameter (the double) contains the bill total, and the second parameter (the String) contains a grade of the service received (A, B, C, D, or F). The tip percent is based on the grade as follows: A: 17%, B: 15% C: 12% D: 7% F: 0%

The calculateTip method should return a double containing the tipAmount that it calculated.

In the ***main*** method, prompt for a bill amount and display a suggested tip for each of the possible grades of service A, B, C, D, and F. Use printf formatting codes to align the values of the table. After the table is displayed, prompt for a grade and calculate and display the tip amount and bill total.

**Sample Output**:

```
> java TipCalculator
What is the amount of the bill? 50

     Grade         Tip
         A        8.50
         B        7.50
         C        6.00
         D        3.50
         F        0.00

Grade the service A, B, C, D, or F : B

Pay $57.50 (Tip: $ 7.50)
```

HINTS:

--- Use printf statements with % formatting codes to mimic the output shown above.
--- A boolean equals operator (==) will not work with two String variables containing the same contents. Instead you must use the String class's .equals() method.

## Spinner Game! (40 points)


Create a class called SpinnerGame which runs a simulation of two people playing a game with a spinner.

The game itself is simple: Spin a wheel which contains the following eight values:

+5 +3 +2 +1 ---1 ---2 ---3 ---5

Player 1 will take a turn by spinning the wheel. The number that the wheel stops on will become that player's score. Player 2 then spins the wheel and earns that score. Play alternates back and forth, where the new spun value is added to the player's total score. Play continues until one of the players reaches a score of 10 or greater. To be fair, if player 1 exceeds 10, player 2 gets one more spin. After the last spin, the player with the highest score greater than 10 is the winner. It's also possible for the game to end in a tie.

*Program Requirements*

- The primary flow of the game occurs in the *main* method, including the loop that cycles between each of the players taking a turn.

    - The loop should continue to loop until one of the players has a score that is greater than or equal to 10.
    - Within the loop, call the *spin* method described below.
    - After you exit the loop, call the *displayWinner* method described below.
- Create a method called *spin*, which has no incoming parameters, but that returns an *int* value containing one of the eight possible spin values. Within the *spin* method, create an array containing the eight values. Generate a random number that contains a random selection of one of the eight array index values and return the number contained at that location in the array.
- Create another method called *displayWinner* that has two incoming parameters containing the scores of player 1 and player 2. Compare the two values and print one of the following messages:
    - Player 1 Won!!
    - Player 2 Won!!
    - Tie!!

**Sample Output**:

```
> java SpinnerGame
Player 1 spun 2 and now has a score of 2
Player 2 spun -5 and now has a score of -5
Player 1 spun -2 and now has a score of 0
Player 2 spun -2 and now has a score of -7
Player 1 spun -3 and now has a score of -3
Player 2 spun 3 and now has a score of -4
Player 1 spun -2 and now has a score of -5
Player 2 spun -1 and now has a score of -5
Player 1 spun -2 and now has a score of -7
Player 2 spun -3 and now has a score of -8
Player 1 spun 5 and now has a score of -2
Player 2 spun -3 and now has a score of -11
Player 1 spun 2 and now has a score of 0
Player 2 spun -1 and now has a score of -12
Player 1 spun 5 and now has a score of 5
Player 2 spun -3 and now has a score of -15
Player 1 spun 5 and now has a score of 10
Player 2 spun 5 and now has a score of -10
Player 1 Won!!

> java SpinnerGame
Player 1 spun 3 and now has a score of 3
Player 2 spun 3 and now has a score of 3
Player 1 spun -3 and now has a score of 0
Player 2 spun 1 and now has a score of 4
Player 1 spun -5 and now has a score of -5
Player 2 spun -1 and now has a score of 3
Player 1 spun 1 and now has a score of -4
Player 2 spun 5 and now has a score of 8
Player 1 spun 5 and now has a score of 1
Player 2 spun -1 and now has a score of 7
Player 1 spun 2 and now has a score of 3
Player 2 spun -2 and now has a score of 5
Player 1 spun 3 and now has a score of 6
Player 2 spun 2 and now has a score of 7
Player 1 spun -1 and now has a score of 5
Player 2 spun 2 and now has a score of 9
Player 1 spun 5 and now has a score of 10
Player 2 spun 2 and now has a score of 11
Player 2 Won!!
```

**NOTES**:
Each program should include comments that explain what each block of code is doing. Additionally, the programs should compile without errors, and run with the results described in the exercise. The following deductions will be made from each exercise if any of the following is incorrect or missing:

- Proper formatting [5 points]
- Proper names for classes and variables [5 points]
- Comments [5 point]
- Program doesn't compile [ 10 points]
- Source code (java file) missing [ 10 points]
- Executable (class file) missing [10 points]
- Missing array where an array was required [5 points]
- Missing loop where a loop was required [5 points]

---

**These solutions are due by 6:00AM on Saturday, October 11.**